



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/773,292	02/09/2004	Ji-young Choi	Q79267	1354
23373	7590	01/07/2008		
SUGHRUE MION, PLLC 2100 PENNSYLVANIA AVENUE, N.W. SUITE 800 WASHINGTON, DC 20037			EXAMINER KISS, ERIC B	
			ART UNIT 2192	PAPER NUMBER
			MAIL DATE 01/07/2008	DELIVERY MODE PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

# Office Action Summary

Application No.

10/773,292

Applicant(s)

CHOI ET AL.

Examiner

Eric B. Kiss

Art Unit

2192

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

## Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

## Status

- 1) ☒ Responsive to communication(s) filed on 12 December 2007.
- 2a) ☐ This action is FINAL. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

## Disposition of Claims

- 4) ☒ Claim(s) 1-6 and 8-11 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-6 and 8-11 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

## Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

## Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- \* See the attached detailed Office action for a list of the certified copies not received.

## Attachment(s)

- ☐ Notice of References Cited (PTO-892)
- ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- ☐ Information Disclosure Statement(s) (PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_
- ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_
- ☐ Notice of Informal Patent Application
- ☐ Other: \_\_\_\_\_

## DETAILED ACTION

### *Continued Examination Under 37 CFR 1.114*

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on December 12, 2007, has been entered. Claims 1-6 and 8-11 are pending.

### *Response to Arguments*

2. Applicant's arguments filed December 12, 2007, have been fully considered but they are not persuasive.

The examiner submits that *Blais* suggests that the runtime data memory and the cache memory may be physically separate. Specifically, in col. 6, lines 30-43, *Blais* discloses a virtual addressing mechanism that allows programs to behave as if they only have access to a large, single storage entity instead of access to multiple, smaller storage entities. Further, *Blais* notes, "[W]hile data 121, operating system 122, class file 123, cache 126, and class processing mechanism 129 are shown to reside in main memory 120, those skilled in the art will recognize that these items are not necessarily all completely contained in main memory 120 at the same time. It should also be noted that the term 'memory' is used herein to generically refer to the entire virtual memory of computer system 100, and may include the virtual memory of other computer systems coupled to computer system 100." From this disclosure, it would have been obvious to one of ordinary skill in the art at the time the invention was made to implement the

first memory unit and second unit as physically separate units under a virtual addressing mechanism consistent with the suggestion of *Blais*, as being able to do so would allow greater flexibility in design while still yielding the same predictable results.

***Claim Rejections - 35 USC § 112***

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. Claims 1-6 and 8-11 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claims 1-6 and 8-11 contain the trademark/trade name JAVA. Where a trademark or trade name is used in a claim as a limitation to identify or describe a particular material or product, the claim does not comply with the requirements of 35 U.S.C. 112, second paragraph. See *Ex parte Simpson*, 218 USPQ 1020 (Bd. App. 1982). The claim scope is uncertain since the trademark or trade name cannot be used properly to identify any particular material or product. A trademark or trade name is used to identify a source of goods, and not the goods themselves. Thus, a trademark or trade name does not identify or describe the goods associated with the trademark or trade name. In the present case, the trademark/trade name is used to identify/describe a particular computer programming language and, accordingly, the identification/description is indefinite.

***Claim Rejections - 35 USC § 103***

5. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

6. Claims **1, 3, 6, 9** and **10** are rejected under 35 U.S.C. 103(a) as being unpatentable over Blais et al., US 7,065,743 (hereinafter **Blais**) in view of Sauntry et al., US6,349,344 (hereinafter **Sauntry**).

In regard to claim **1, Blais**) discloses:

- *“A system for shortening a class loading process in a Java program, comprising: a class loader unit for loading Java program class files from an auxiliary memory, performing...and initialization processes and generating runtime data...”* (E.g., see Figure 4 & Column 7, lines 60-Column 8, line 13), wherein the process begins upon class loading, wherein initialization processes and runtime information is generated.
- *“...a first memory unit for maintaining the runtime data generated by the class loader unit in an accessible state...”* (E.g., see Figure 1, element 120 & Column 6, lines 30-40), wherein the main memory stores the necessary runtime information and data that processor 110 may access.
- *“...a second memory unit for storing the runtime data, which have been loaded into the first memory unit in the accessible state ...”* (E.g., see Figure 3 (127) & Column 7, lines 19-31), wherein the cache is a second memory unit storing the already processed runtime data (analyzed program).
- *“...a runtime data search unit for loading the runtime data, which have been stored in the second memory unit ... into the first memory unit upon the request of the class loader unit...”* (E.g., see Figure 5 & Column 8, lines 42-44), wherein the runtime data search unit (cache processing mechanism, see

Figure 1, element 129) determines if a class is stored in the runtime data (cache 126).

- "...and an execution unit for executing the runtime data that have been loaded into the first memory unit in the accessible state..." (E.g., see Figure 1 (110) & Column 5, lines 32-35), wherein a processor for executing the runtime data is disclosed.
- "...wherein said first memory unit and said second memory unit are separate." (E.g., see Figure 1 (126) & Column 6, lines 30-43 + 17-21), wherein a processor 110 for executing the instructions enables the class processing mechanism 129 to retrieve the analyzed class data from the separate cache 126.

**Blais** further suggests that the runtime data memory and the cache memory may be physically separate. Specifically, in col. 6, lines 30-43, *Blais* discloses a virtual addressing mechanism that allows programs to behave as if they only have access to a large, single storage entity instead of access to multiple, smaller storage entities. Further, *Blais* notes, "[W]hile data 121, operating system 122, class file 123, cache 126, and class processing mechanism 129 are shown to reside in main memory 120, those skilled in the art will recognize that these items are not necessarily all completely contained in main memory 120 at the same time. It should also be noted that the term 'memory' is used herein to generically refer to the entire virtual memory of computer system 100, and may include the virtual memory of other computer systems coupled to computer system 100." From this disclosure, it would have been obvious to one of ordinary skill in the art at the time the invention was made to implement the first memory unit and second unit

as physically separate units under a virtual addressing mechanism consistent with the suggestion of *Blais*, as being able to do so would allow greater flexibility in design while still yielding the same predictable results.

But **Blais** does not expressly disclose “linking” or “storing in a form of images”.

However, **Sauntry** discloses:

- “...*linking*...” (E.g., see Figure 3b (354) & Column 9, lines 54-60), wherein references are loaded.
- “...*in the form of images*...” (E.g., see Figure 1 (110) & Column 3, lines 2-3), wherein generated class files are stored as a run-time image.

**Blais** and **Sauntry** are analogous art because they are both concerned with the same field of endeavor, namely, managing java class files. Therefore, at the time the invention was made, it would have been obvious to a person of ordinary skill in the art to combine **Sauntrys**’ class file storing method with **Blais**’ class file retrieval method. The motivation was provided by **Blais** in developing a method to overcome the shortcomings of java class file loading and parsing at run-time by a Java virtual machine so that Java programs may realistically be more able to run on memory-constrained platforms (see Column 2, lines 58-63).

In regard to claim 3, the rejections of base claim 1 are incorporated. Furthermore, **Blais** discloses:

- “...*the runtime data search unit causes the runtime data generated by the class loader unit to be stored in the second memory unit*...” (E.g., see Figure 5 (532) & Column 8, lines 64-66), wherein the program information is stored in the cache.

Claim 6 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Blais**.

In regard to claim 6, **Blais** discloses:

- "...a runtime data search unit to search runtime data necessary for execution of the Java program, by a class loader unit;; ..." (E.g., see Figure 5 & Column 8, lines 42-44), wherein the runtime data search unit (cache processing mechanism, see Figure 1, element 129) determines if a class is stored in the runtime data (cache 126).
- "...searching the requested runtime data for the Java program by the runtime data search unit ..." (E.g., see Figure 5 (522) & Column 8, lines 56-58), wherein the cache is searched.
- "...transmitting the searched runtime data to a first memory unit; and executing the runtime data transmitted to the first memory unit..." (E.g., see Figure 5 (526) & Column 8, lines 58-60), wherein the analyzed program information is read from the cache entry (step 526) and used (step 528).

**Blais** further suggests that the runtime data memory and the cache memory may be physically separate. Specifically, in col. 6, lines 30-43, *Blais* discloses a virtual addressing mechanism that allows programs to behave as if they only have access to a large, single storage entity instead of access to multiple, smaller storage entities. Further, *Blais* notes, "[W]hile data 121, operating system 122, class file 123, cache 126, and class processing mechanism 129 are shown to reside in main memory 120, those skilled in the art will recognize that these items are not necessarily all completely contained in main memory 120 at the same time. It should also be noted that the term 'memory' is used herein to generically refer to the entire virtual memory of



computer system 100, and may include the virtual memory of other computer systems coupled to computer system 100.” From this disclosure, it would have been obvious to one of ordinary skill in the art at the time the invention was made to implement the first memory unit and second unit as physically separate units under a virtual addressing mechanism consistent with the suggestion of *Blais*, as being able to do so would allow greater flexibility in design while still yielding the same predictable results.

But **Blais** does not expressly disclose “requesting” the cache processing mechanism 129 (runtime search unit) to search. However, it would have been obvious to one of ordinary skill in the art, at the time the invention was made, that a needed class is equivalent to requesting the class, as it would be necessary to request a needed class in order to receive it. See claim 1 for the remaining limitations.

In regard to claim 9, the rejections of base claim 6 are incorporated. Furthermore, **Blais** discloses:

- “...if it is determined from search results of the requested runtime data for the Java program that there are no relevant runtime data, loading Java program class files from an auxiliary memory...” (E.g., see Column 3, lines 20-23), wherein if there is no entry in the cache corresponding to the class the program information is analyzed and saved in a cache entry for future use.

See claim 1 for the remaining limitations.

In regard to claim 10, the rejections of base claim 9 are incorporated. Furthermore, **Blais** discloses:

- "...the step of storing the generated runtime data..." (E.g., see Figure 4 & Column 9, lines 31-36), wherein.

But **Blais** does not expressly disclose "*performed after the step of executing the runtime data transmitted to the first memory unit*". However, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to perform the storing step after execution rather than before as disclosed by **Blais**. The motivation to do so is that it is old and well known in the art to execute code steps in different order, particularly when the result (e.g., storing and executing) is the same and not effected by the order of execution. Accordingly, **Blais'** disclosure of storing and then executing the runtime information produces the same result as executing and then storing. Thus, storing after execution would have been obvious to one of ordinary skill in the art.

7. Claims **2, 4, 5, 8** and **11** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Blais** in view of **Sauntry** and further in view of Rodriguez et al., US 6,725,241 (hereinafter **Rodriguez**).

In regard to claim **2**, the rejections of base claim **1** are incorporated. But **Blais** and **Sauntry** do not expressly disclose "...a garbage collector unit for collecting space unused in the first memory unit and allowing the unused space to be used again." However, **Rodriguez** discloses:

- "...a garbage collector unit for collecting space unused in the first memory unit and allowing the unused space to be used again." (E.g., see Figure 4 &

Column 3, lines 45-67), wherein a garbage collection unit for allowing unused space to be used again is disclosed.

**Blais, Sauntry** and **Rodriguez** are analogous art because they are both concerned with the same field of endeavor, namely, managing java virtual machine memory. Therefore, at the time the invention was made, it would have been obvious to a person of ordinary skill in the art to combine **Sauntrys'** garbage collector via least recently used method with **Blais and Sauntrys'** class file retrieval method. The motivation was provided by **Blais'** teaching to overcome the shortcomings of java class file loading and parsing at run-time by a Java virtual machine so that Java programs may realistically be more able to run on memory-constrained platforms (see Column 2, lines 58-63). Further motivation was provided by **Rodriguez's** disclosure of freeing memory in native method stacks (see Figure 2, 212 & Column 4, lines 30-33).

In regard to claim 4, the rejections of base claim 1 are incorporated. But **Blais** and **Sauntry** do not expressly disclose "...by using a least recently used (LRU) method." However, **Rodriguez** discloses:

- "...by using a least recently used (LRU) method." (E.g., see Figure 4 & Column 3, lines 45-67), wherein a garbage collection unit for allowing unused space to be used again via the least recently used method is disclosed.

In regard to claims 5, 8 and 11, see claim 4.

### ***Conclusion***

8. Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Eric B. Kiss whose telephone number is (571) 272-3699. The

Application/Control Number:  
10/773,292  
Art Unit: 2192

Page 11

Examiner can normally be reached on Tue. - Fri., 7:00 am - 4:30 pm. The Examiner can also be reached on alternate Mondays.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Tuan Dam, can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Any inquiry of a general nature should be directed to the TC 2100 Group receptionist:  
571-272-2100.



Eric B. Kiss  
Primary Patent Examiner  
January 4, 2008